Doc Code: AP.PRE.REQ

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Docket Number (Optional): 10019982-1

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)]

on 12/06/06

Signature [illegible]

Typed or printed name Desiree Reardon

Application Number: 10/017,342	Filed: 12/13/01
First Named Inventor: Robert HUNDT	
Art Unit: 2193	Examiner: Mitchell, J. D.

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)

☒ attorney or agent of record. 35,398
Registration number

☐ attorney or agent acting under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 ____________

[illegible]
Signature

John P. Wagner Jr.
Typed or printed name

408-938-9060
Telephone number

12/6/06
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required.
Submit multiple forms if more than one signature is required, see below*.

☐ *Total of ________ forms are submitted.

REMARKS ACCOMPANYING PRE-APPEAL BRIEF REQUEST FOR REVIEW

In response to the Final Office Action dated September 6, 2006, Applicants respectfully request a review of the final rejection in the above-identified application. Applicants respectfully submit that the Examiner's rejections of the Claims are improper. Claims 1-15 are rejected under 35 U.S.C. §103(a) as being unpatentable over by U.S. patent no. 6,189,141 by Benitez et al. (referred to hereinafter as "Benitez") in view of "Unix Programming Frequently Asked Questions -1. Process Control" (referred to hereinafter as "Unix").

KEY CLAIM LIMITATIONS THAT ARE NOT MET BY THE CITED REFERENCES

Neither Benitez nor Unix teach or suggest "modifying text segment portions selected from said child process resulting in an uninstrumented code space that is an uninstrumented version of said instrumented code space...provided an instruction pointer resides in said instrumented code space, updating said instruction pointer to point to said uninstrumented code space," as recited by Claim 1. Independent Claims 6 and 11 should be allowable for similar reasons that Claim 1 should be allowable.

Claim Limitations Having To Do With Modifying Text Segment Portions Selected From Said Child Process Resulting In An Uninstrumented Code Space That Is An Uninstrumented Version Of Said Instrumented Code Space...Prov8ided An Instruction Pointer Resides In Said Instrumented Code Space, Updating Said Instruction Pointer To Point To Said Uninstrumented Code Space Are Not Met By The Cited References

Benitez teaches a method and a system for optiziming code by identifying and translating hot blocks. Benitez also teaches a method and system for reducing unnecessary translations and optimizations to increase the speed of executing code. This is accomplished by identifying traces within the original code and determining the frequency with which those traces are executed or emulated. If the traces are executed frequently, the traces are "hot traces" and are candidates for translation/optimization. However, if the traces are executed infrequently, the traces are "cold traces" and are not candidates for translation/optimization. The frequency with which the traces are executed is tracked and compared to a threshold to determine which traces are "hot" and which traces are "cold."

As is well known in the art, a jump instruction may cause execution to jump to a second instruction by specifying the target address of the second instruction. Assume for the sake of illustrating Benitez that at the beginning of a second portion of original instructions includes a jump instruction that causes execution to jump to the target address of another instruction in a first portion of original instructions. Also assume for the sake of illustrating Benitez that these two portions of original instructions are identified as hot blocks and are translated into respective first and second translated hot blocks. The target address has changed due to the translations and therefore the target address that the jump instruction would jump to would need to be changed to correctly identify the corresponding target address in the first translated hot block. Benitez discusses using a backpatcher at Col. 3 lines 18-24 so that the jump instruction in the second translated hot block correctly causes execution to jump to the new target address in the first translated hot block.

Now assume for the sake of illustrating Benitez that the first translated hot block is identified as "cold" and therefore the first portion of original code will be executed instead of the first translated hot block. In this case, the target address specified by the jump instruction in the second translated hot block will need to be changed back. Benitez discuses using the backpatcher at Col. 29 lines 19-24 to correct the target address specified by the jump instruction.

Claim 1 recites, "modifying text segment portions selected from said child process resulting in an uninstrumented code space that is an uninstrumented version of said instrumented code space." In contrast, Benitez's original instructions appear to always be present and the instrumented hot blocks are instrumented versions of portions of the original instructions. For example, refer to Col. 14 lines 44-46 which states, "...original instructions generally are emulated if they are not identified as part of a hot block, and therefore translated." Then referring to Col. 9 lines 51-56, the translated hot blocks are "removed" from memory when they are identified as being "cold." <u>Further note that the Office Action does not provide any arguments against "...resulting in an uninstrumented code space that is an uninstrumented version of said instrumented code space."</u>

It appears that the Office Action is asserting that Benitez's translated hot blocks are analogous to Claim 1's "instrumented code space." Since Benitez teaches removing translated hot blocks from memory, Benitez cannot teach "provided <u>an instruction pointer resides in said instrumented code space, updating said instruction pointer</u> to point to said uninstrumented code space," (emphasis added) as Claim 1 recites.

The Response to Arguments section of the Office Action dated September 6, 2006 did not rebuttle Applicants' arguments with respect to "modifying text segment portions selected from said child process resulting in an uninstrumented code space that is an uninstrumented version of said instrumented code space." Further, the Office Action does not rebuttle Applicants' arguments with respect to "resulting in an uninstrumented code space that is an uninstrumented version of said instrumented code space" anywhere.

With respect to Applicants' arguments concerning "provided an instruction pointer resides in said instrumented code space, updating said instruction pointer to point to said uninstrumented code space ", the Response to Arguments section of the Office Action dated September 6, 2006 stated "Removing a 'hot trace' that has 'gone cold' from memory would leave pointers in other 'hot traces' (instrumented code space) which would, after such a removal, but pointing to 'unmapped' instructions in the removed 'hot trace.' It is these instruction pointers residing in instrumented coded space which Benitez updates to point to said uninstrumented code space…" It appears that the Office Action is asserting that Benitez' "hot trace" that has gone cold and Benitez' "other 'hot traces'" which haven't gone cold are both analogous to Claim 1's instrumented code space. Applicants wish to point out that Claim 1 recites "uninstrmented code space that is an uninstrumented version of said instrumented code space…unmapping said instrumented code space…provided an instruction pointer resides in said instrumented code space, updating said instruction pointer to point to said uninstrumented code space." Note that Claim 1's "instruction pointer" resides in the instrumented code space that was unmapped. In contrast, assuming solely for the sake of argument the Office Action's analogies between Benitez and Claim 1, Benitez' instruction pointer resides in the "other 'hot space'" which is still mapped. Therefore, Benitez cannot teach or suggest, among other things, "provided an instruction pointer resides in said instrumented code space, updating said instruction pointer to point to said uninstremented code space," as recited by Claim 1.

Unix does not remedy the shortcomings in Benitez in that Unix does not teach or suggest "modifying text segment portions selected from said child process resulting in an uninstrumented code space that is an uninstrumented version of said instrumented code space...provided an instruction pointer resides in said instrumented code space, updating said instruction pointer to point to said uninstrumented code space," as recited by Claim 1. In fact, the Office Action does not even assert that Unix teaches "modifying text segment portions selected from said child process resulting in an uninstrumented code space that is an uninstrumented version of said instrumented code space...provided an instruction pointer resides in said instrumented code space, updating said instruction pointer to point to said uninstrumented code space," as recited by Claim 1.

In summary, Applicants respectfully submit that the Examiner's rejections of the Claims are improper as key limitations of Claim 1 (from which Claims 2-5 depend), Claim 6 (from which Claims 7-10 depend) and Claim 11 (from which Claims 12-15 depend) are neither taught nor suggested by Benitez nor Unix, alone or in combination, as outlined above. Moreover, because key limitations of independent Claims 1, 6 and 11 are neither taught nor suggested by Benitez or Unix, alone or in combination, Applicants respectfully submit that the rejection of the Claims 1, 6, and 11 under 35 U.S.C. §103(a) as being unpatentable over Benitez in view of Unix are improper and should be reversed.